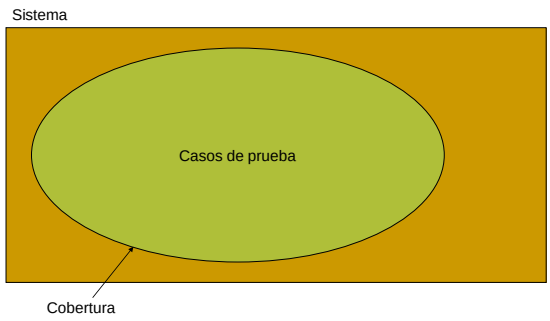
Casos de prueba

Técnicas de desarrollo

Cobertura



Caso de prueba

Condiciones

- Pre-condiciones
- Post-condiciones

Input

Output esperado

Procedimiento de prueba

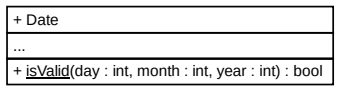
- Instrucciones detalladas de cómo ejecutar un conjunto de casos de prueba.

Niveles de prueba

- Definimos arbitrariamente:
  - Unidad -> Test Item = Clase
  - Integración -> Test Item = Service Package
  - Sistema -> Test Item = Use Case

Casos de prueba - Métodos

Tomemos como ejemplo de unidad esta clase....




```
public class Date {
      public static boolean
      isValid(int day, int month, int year)
}
```


Casos de prueba - Métodos

1.- Análisis de Dominio

```
isValid(int day, int month, int year)
```

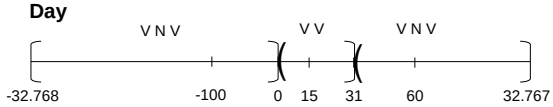
int en un entero de 16 bits (2^{16} = 65.536)
Entonces...

-32.768 <= day <= 32.767
-32.768 <= month <= 32.767
-32.768 <= year <= 32.767

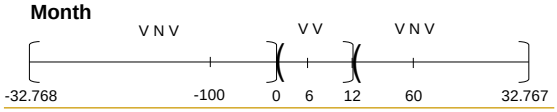
Casos de prueba - Métodos

2.- Equivalence Partitioning
(Particiones Equivalentes)

Day



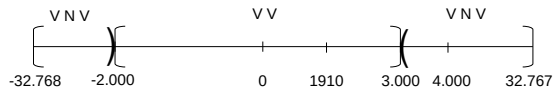
Month



Casos de prueba - Métodos

2.- Equivalence Partitioning (cont.)
(Particiones Equivalentes)

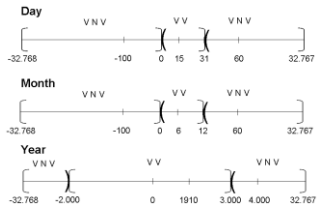
Year



Casos de prueba - Métodos

2.- Equivalence Partitioning (cont.)
(Particiones Equivalentes)

		CP 1	CP 2	CP 3
Input	Day	15	-100	60
	Month	6	-100	60
	Year	1910	-3000	4000
Output Esperado		True	False	False



Casos de prueba - Métodos

3.- Boundary Value Analysis
(Análisis de valores en el borde del dominio)

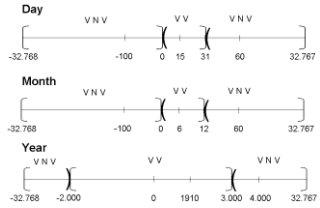
A. Para cada partición equivalente utilizar dos valores lo más cercanos posible al borde del dominio.

B. Buscar valores de entrada que generen valores de salida en el límite del dominio.

Casos de prueba - Métodos

3.- Boundary Value Analysis (cont.)
(Análisis de valores en el borde del dominio)

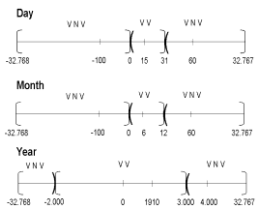
		CP 1	CP 2	CP 3
Input	Day	~~15~~ 1	~~-100~~ 0	~~60~~ 32
	Month	~~8~~ 1	~~-100~~ 0	~~60~~ 13
	Year	1910 -2000	~~-3000~~ -2001	~~4000~~ 3001
Output Esperado		True	False	False



Casos de prueba - Métodos

3.- Boundary Value Analysis (cont.)
(Análisis de valores en el borde del dominio)

		CP 1	CP 2	CP 3	CP 4
Input	Day	~~15~~ 1	~~-100~~ 0	~~60~~ 32	31
	Month	~~8~~ 1	~~-100~~ 0	~~60~~ 13	12
	Year	1910 -2000	~~-3000~~ -2001	~~4000~~ 3001	3000
Output Esperado		True	False	False	True



Casos de prueba - Métodos

4.- Error Guessing

Teniendo conocimiento del dominio del problema:

A. Hacer una lista de posible errores.

B. Crear casos de prueba para cada caso de la lista.

Casos de prueba - Métodos

4.- Error Guessing (cont.)

		CP 1	CP 2	CP 3	CP 4	CP 5	CP 6
Input	Day	1	0	32	31	29	29
	Month	1	0	13	12	2	2
	Year	-2000	-2001	3001	3000	2009	2008
Output Esperado		True	False	False	True	False	True

Casos de prueba - Métodos

4.- Exploratory Testing

Sin conocimiento del dominio del problema:

A. Investigar acerca de:
   - El producto
   - El mercado
   - Debilidades del producto
   - Formas en las que podría fallar el producto
   - Formas en las que se lo podría probar

B. Probar el producto

Casos de prueba - Métodos

4.- Exploratory Testing (cont.)

C. Reportar errores.
D. Crear nuevas pruebas en base a lo aprendido.
E. Volver a empezar.

Casos de prueba – Técnicas

- Stress Testing
  - El foco es hacer cosas que son tan difíciles de manejar para el sistema que eventualmente fallará.
  - Puede servir para encontrar el límite de operación en algún eje.
  - Ejemplo: ¿Cuántas transacciones por minuto puede procesar antes de fallar.

Casos de prueba – Técnicas

- Regression testing
  - Se ejecuta luego de un cambio
  - Intenta identificar si algo se rompió inesperadamente como consecuencia indirecta del cambio.
  - Generalmente es automático ya que de otra forma sería demasiado caro.

Casos de prueba – Técnicas

- Scenario testing
  - Reproducir escenarios reales de uso, involucrando secuencias de casos de uso, y documentarlos en casos de prueba.
  - Tener siempre en cuenta los flujos alternativos.
  - También se debe pensar en flujos que no necesariamente estén documentados en los casos de uso pero que los usuarios muy probablemente ejecutarán.
  - No hay que utilizarlo sólo ya que nos estaríamos perdiendo la funcionalidad fuera de los usos frecuentes o esperados.

Casos de prueba – Técnicas

- Random testing
  - La estrategia es diseñada por un humano.
  - Los datos que la implementan son generados estocasticamente, dentro de ciertos parámetros, por una herramienta.
  - La efectividad está en la cantidad y NO en la calidad

Casos de prueba – Técnicas

- Specification Based Test
  - La idea: verificar cada afirmación hecha en las especificaciones.
  - Más allá de las especificaciones formales es una buena idea basarse en:
    - Manual del usuario
    - Pedidos de cambio documentados

Casos de prueba – Técnicas

- Operation Test
  - Se hace en el entorno de producción.
  - Se pone a funcionar el sistema en su entorno real y se toman métricas de fallas. Por ejemplo: “Mean Time To Failure” (Tiempo promedio entre fallas).
  - Suele utilizarse en grandes sistemas en donde es muy costoso crear el “Test Bed”.

Casos de prueba – Técnicas

- Full Scale Test
  - Se prueba el sistema al ***máximo*** de su capacidad, no más allá. Si no, sería un Stress Test.
  - Por ejemplo:
    - Probar una tabla de base de datos con la máxima cantidad de registros que soporta según especificación.
    - Probar un sistema con la máxima cantidad de transacciones simultáneas que soporta.

Casos de prueba – Técnicas

- Ergonomic Test
  - Prueba de los objetivos de usabilidad del sistema.
  - Por ejemplo:
    - Curva de aprendizaje.
    - Velocidad de data entry.

Casos de prueba – Técnicas

- Configuration Tests
  - Se realizan pruebas sobre diferentes configuraciones posibles.
  - Se usa en sistemas multiplataforma.

Cobertura